access memory (RAM) chips (108); disk controller devices 110 (for example, Integrated Device Electronics (IDE) or Small Computer System Interface (SCSI) controllers, connected to a floppy disk drive, a hard disk drive, and additional removable media drives such as Digital Versatile Disc (DVD) drives); one or more input/output ports (112) (for example, one or more Universal Serial Bus (USB) port controllers, and/or parallel port controllers for connection to printer and so on); an expansion bus 114 for bus connection to external or internal peripheral devices (for example, the Peripheral Component Interconnect (PCI) bus); and other system chips 116 (for example, graphics and sound devices). Examples [[or]] of computers of this type are personal computers (PCs) and workstations. However, the application of the invention to other computing devices such as mainframes, embedded microcomputers in control systems, and Personal Digital Assistants (PDAs) (in which case some of the indicated devices such as disk drive controllers may be absent) is also disclosed herein.

Management of Software

Referring to FIG. 2a, in use, the computer 100 of FIG. 1 runs resident programs comprising operating system kernel 202 (which provides the output routines allowing access by the CPU to the other devices shown in FIG. 1); an operating system user interface or presentation layer 204 (such as X Windows); a middleware layer 206 (providing networking software and protocols such as, for instance, a Transmission Control Protocol/Internet Protocol (TCP/IP) stack) and applications 208a, 208b, which run by making calls to the Application Programming Interface (API) routines forming the operating system kernel 202.

Change(s) applied to document, /J.E.B./ 6/18/2011

Page ~~8, 8th~~ 10, 3rd full paragraph:

The operating systems are not treated equally by the embodiment. Instead, one of the operating systems is selected as the "critical" operating system[[s]] (this will be the real time operating system), and the or each other operating system is treated as a

started by the hardware resource dispatcher, in its allocated memory space, rather than starting itself. The hardware-probing stage of the initialisation sequence is modified, to prevent the critical operating system from accessing the hardware resources which are assigned to other secondary systems. It reads the static hardware allocation table from the hardware resource dispatcher to detect the devices available to it.

Page 13, 1st paragraph:

Trap calls [[2012]] are added to the critical operating system, to detect states and request some actions in response. A trap call here means a call which causes the processor to save the current context (e.g. state of registers) and load a new context. Thus, where virtual memory addressing is used, the address pointers are changed. For example, when the real time operating system 201 reaches an end point (and ceases to require processor resources) control can be passed back to the hardware resource dispatcher, issuing the "idle" trap call, to start the secondary operating system. Many processors have a "halt" instruction. In some cases, only supervisor-level code (e.g. operating systems, not applications) can include such a "halt" instruction. In this embodiment, all the operating systems are rewritten to remove "halt" instructions and replace them with an "idle" routine (e.g. an execution thread) which, when called, issues the "idle" trap call.

Change(s) applied to document, /J.E.B./ 6/18/2011

Page 13, ~~2nd~~ 3rd full paragraph:

Additional "virtual" drivers [[2014]] are added which, to the operating system, appear to provide access to an input/output (I/O) bus, allowing data to be written to the bus. In fact, the virtual bus driver [[2014]] uses memory as a communications medium; it exports some private memory (for input data) and imports memory exported by other systems (for output data). In this way, the operating system 201 (or an application running on the operating system) can pass data to another operating system (or